

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently amended) In a network having a VMS with a plurality of data and voice processing boards performing the same function and a plurality of external servers, a system for unifying multiple connection ports, comprising:

an external input/output controller (EIOC) configured to provide unified network connection point between the plurality of boards and the plurality of external servers by routing messages from any one of the plurality of boards to any one of the external servers and by routing messages from any one of the plurality of servers to any one of the plurality of ~~boards~~boards,

wherein the EIOC checks a header of a message when the message is transferred from one of the plurality of boards of the VMS to the EIOC or from one of the external servers to the EIOC, to determine the destination address of the message.

2. (Original) The system according to claim 1, wherein the EIOC comprises:  
a main interface unit configured to connect with the plurality of boards;

an external interface unit configured to connect with the plurality of external servers; and

a shared memory which is configured so that messages can be exchanged between the main interface unit and the external interface unit through the shared memory.

3. (Original) The system according to claim 2, wherein the main interface unit comprises:

a communication server and communication client configured to communicate with the plurality of boards;

a shared memory transfer unit configured to transfer a message from one of the plurality of boards to the shared memory, and to generate an interrupt signal to inform the external interface unit that a message is stored in the shared memory; and

a shared memory polling unit configured to periodically poll the shared memory to determine if a message is waiting to be sent or delivered.

4. (Original) The system according to claim 3, wherein the main interface unit further comprises a shared memory receiving unit configured to analyze a message read from the shared memory, and configured to route the message to one of the plurality of boards of the VMS based on the results of the analysis.

5. (Original) The system according to claim 4, wherein the main interface unit further comprises:

a process management unit configured to generate and monitor all processes performed by the main interface unit; and

a child block management unit configured to drive a child process to transfer and receive an operation message and state information under the control of the process management unit.

6. (Original) The system according to claim 5, wherein the external interface unit comprises first and second external interface units, each of which is configured to connect to all of the plurality of external servers, and wherein the main interface unit further comprises:

a duplex communication server configured to control a duplex operation of the first and second external interface units; and

a NMS server configured to process an internal operation message and to collect state information.

7. (Original) The system according to claim 2, wherein the external interface unit comprises:

a shared memory polling unit configured to periodically poll the shared memory to determine if a message is waiting to be sent or delivered;

a shared memory receiving unit configured to analyze a message read from the shared memory, and configured to deliver a message to one of the plurality of external servers based on the results of the analysis.

8. (Original) The system according to claim 7, wherein the external interface unit further comprises:

a communication server and communication client configured to communicate with the plurality of external servers; and

a shared memory transfer unit configured to transfer a message from one of the external servers to the shared memory, and configured to generate an interrupt to inform the main interface unit that a message is stored in the shared memory.

9. (Original) The system according to claim 8, wherein the external interface unit further comprises a process management unit configured to generate and monitor all processes performed by the external interface unit.

10. (Original) The system according to claim 2, wherein the main interface unit and the external interface unit inform each other that there is a message to be delivered by generating an interrupt after storing the message in the shared memory, and wherein the main interface unit and the external interface unit recognize that there is a message to be delivered if they receive an interrupt.

11. (Currently amended) A method of unifying multiple connection ports in a network that includes a VMS having a plurality of data and call processing boards, and a plurality of external servers, comprising:

a forward message transfer ~~step that comprises~~comprising:

checking the header of a message with an external input/output controller (EIOC), when the message is transferred from one of the plurality of boards of the VMS to the EIOC, to determine the destination address of the message, and

transferring the message to one of the plurality of external servers based on the determined destination address; and

a backward message transfer ~~step that comprises~~comprising:

checking the header of a message with the external input/output controller (EIOC), when the message is transferred from one of the external servers to the EIOC, to determine the destination address of the message, and

transferring the message to one of the plurality of boards of the VMS based on the determined destination address.

12. (Currently amended) The method of claim 11, wherein the forward message transfer ~~step~~ further comprises:

receiving a message from one of the boards of the VMS in a main interface unit of the EIOC;

recording the message in a shared memory of the EIOC;

generating an interrupt with the main interface unit to inform an external interface unit of the EIOC that a message is stored in the shared memory;

detecting the interrupt with the external interface unit;

reading the message from the shared memory with the external interface unit;

checking a header of the message with the external interface unit to determine a destination address of the message; and

transferring the message from the external interface unit to one of the plurality of external servers based on the determined destination address.

13. (Currently amended) The method according to claim 12, wherein in the receiving ~~step~~, the board transferring the message also adds an internal message header to the message so that the message can be routed to a desired destination external server.

14. (Original) The method according to claim 12, wherein the board transfers a message file to the EIOC by dividing the message into a plurality of data packets.

15. (Original) The method according to claim 14, wherein when a message is transferred as a plurality of data packets, the data packets are temporarily stored in a local memory unit of the main interface unit until the entire message is received, and wherein the entire message is then copied to the shared memory.

16. (Currently amended) The method according to claim 12, wherein the recording ~~step~~ comprises checking a network state and storing the message in a region of the shared memory so that it is transferred to the next available external interface unit of a duplex external interface unit.

17. (Currently amended) The method according to claim 12, wherein the reading ~~step~~ comprises copying a file from the shared memory into a local memory of the external interface unit, and then transferring a response to the main interface unit.

18. (Currently amended) The method according to claim 11, wherein the backward message transfer ~~step~~ comprises:

receiving a message from one of the external servers with an external interface unit of the EIOC;

storing the received message in a shared memory of the EIOC;

generating an interrupt with the external interface unit to inform a main interface unit that the message is stored in the shared memory;

detecting the interrupt with the main interface unit;

reading the message from the shared memory with the main interface unit; and

routing the message to one of the plurality of boards of the VMS based on destination information stored in the message.

19. (Currently amended) The method according to claim 18, wherein the reading ~~step~~ comprises storing the message read from the shared memory in a local memory of the main interface unit, and wherein the routing ~~step~~ comprises:

Serial No. **09/695,259**

Docket No. **P-0138**

Amdt. dated May 18, 2006

Reply to Office Action of February 22, 2006

if the message stored in the local memory is an overload-related message, transferring the message according to an overload information or broadcast transferring the message;

if the message is a response message, routing the message to a corresponding board of the VMS based on destination information stored in the message;

if the message is a message added or deleted to/from a mail box, routing the message to a call processing board of the VMS; and

if the message is some other type of message, routing the message to a corresponding board of the VMS using a mail box information stored in the message.